

# Why Framework?

From Manual PHP to Laravel

# What You've Built So Far

## Manual PHP REST API Structure

```
api/  
├── index.php          # Manual routing  
├── handlers.php       # CRUD functions  
├── models/Student.php # Manual model  
└── data/students.json # File storage
```

**200+ lines of code** for basic CRUD operations!

# The Problems with Manual PHP

## 1. Repetitive Code

```
// You write this everywhere  
header('Content-Type: application/json');  
$input = json_decode(file_get_contents('php://input'), true);
```

## 2. Manual Routing

```
// Complex routing logic
$path = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
$segments = explode('/', $path);
switch ($method) { ... }
```

## 3. No Structure

- Files scattered everywhere
- Hard to maintain as the project grows
- No consistent patterns

# What is a Framework?

A **framework** offers a **ready-made foundation** with built-in tools and patterns, so developers can focus on solving **business logic** instead of handling the **low-level plumbing**.

## Restaurant Analogy

- **Manual PHP** = Building a restaurant from scratch (plumbing, electricity, kitchen...)
- **Framework** = Renting a fully-equipped restaurant (focus on cooking and serving)

# Laravel: The Popular PHP Framework

## Laravel: Making PHP Easier

- **Framework built on PHP**

Laravel takes raw PHP and wraps it in a **clean, modern structure**.

- **Simplifies common tasks**

- Routing (URLs → functions)
- Authentication & security
- Database queries with Eloquent ORM
- Blade template engine for HTML/PHP mix

## Used by Industry Giants

- **Laravel powers millions of websites**
- Disney, 9GAG, Pfizer, BBC, Toyota
- GitHub: 70,000+ stars

## Why Laravel?

- **Elegant syntax** - Clean, readable code
- **Powerful features** - Built-in tools for everything
- **Great documentation** - Excellent learning resources
- **Large community** - Easy to find help

# Laravel vs Your Manual Code

## Manual PHP (What You Built)

```
// 15 lines just for routing!
$method = $_SERVER['REQUEST_METHOD'];
$path = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
$segments = explode('/', $path);
$resource = $segments[0] ?? '';

switch ($method) {
    case 'GET':
        if ($student_id) { get_student($student_id); }
        else { get_all_students(); }
        break;
    // ... more cases
}
```



## Laravel (Same functionality)

```
// Just 3 lines!  
Route::get('/students', [StudentController::class, 'index']);  
Route::get('/students/{id}', [StudentController::class, 'show']);  
Route::post('/students', [StudentController::class, 'store']);
```

Laravel handles all the complexity for you!

# Framework Benefits

## 1. Less Code, More Features

- Your manual API: 200+ lines
- Laravel equivalent: 50 lines
- More features, fewer bugs

## 2. Industry Standards

- Professional code structure
- Security best practices built in
- Scalable architecture

## 3. Career Ready

- Companies expect framework knowledge
- Higher salary potential
- Faster development speed

# Laravel Disadvantages

- Learning curve
- Performance overhead due to layers of abstractions
- Less flexibility to follow Laravel rules
- Version upgrades may cause conflicts
- Possible hosting/deployment issues
- Hidden complexity

# Real-World Impact

## Development Speed

- **Manual PHP:** 2 weeks for basic CRUD
- **Laravel:** 2 days for the same features

## Code Quality







- **Manual PHP:** Custom patterns, potential bugs
- **Laravel:** Proven patterns, tested code

## Maintenance

- **Manual PHP:** Hard to modify and extend
- **Laravel:** Easy to update and scale

# Laravel's Magic

## What Laravel Provides Automatically:

-  **Routing** - Clean URL handling
-  **Database ORM** - No more manual SQL
-  **Security** - CSRF, validation, authentication
-  **Testing** - Built-in testing tools
-  **API Development** - JSON responses, rate limiting
-  **File Structure** - Organized, maintainable code

## Key Takeaway

You've learned to build with **raw materials**. Now you'll learn to create with **power tools**.

**Manual PHP taught you the fundamentals.**

**Laravel will make you productive and professional.**

**Ready to see the magic? Let's install Laravel!**